Static Aeroelastic Analysis with an Inviscid Cartesian Method

David L. Rodriguez*
Science & Technology Corp., Moffett Field, CA 94305

Michael J. Aftosmis†
NASA Ames Research Center, Moffett Field, CA 94305

Marian Nemec‡
Science & Technology Corp., Moffett Field, CA 94305

Stephen C. Smith§
Zee Aero, Mountain View, CA

An embedded-boundary Cartesian-mesh flow solver is coupled with a three degree-of-freedom structural model to perform static, aeroelastic analysis of complex aircraft geometries. The approach solves the complete system of aero-structural equations using a modular, loosely-coupled strategy which allows the lower-fidelity structural model to deform the high-fidelity CFD model. The approach uses an open-source, 3-D discrete-geometry engine to deform a triangulated surface geometry according to the shape predicted by the structural model under the computed aerodynamic loads. The deformation scheme is capable of modeling large deflections and is applicable to the design of modern, very-flexible transport wings. The interface is modular so that aerodynamic or structural analysis methods can be easily swapped or enhanced. *This extended abstract includes a brief description of the architecture, along with some preliminary validation of underlying assumptions and early results on a generic 3D transport model. The final paper will present more concrete cases and validation of the approach. Preliminary results demonstrate convergence of the complete aero-structural system and investigate the accuracy of the approximations used in the formulation of the structural model.*

I. Introduction

In contrast to the relatively rigid aluminum wings of the past seven decades, modern composite wings are significantly more flexible. For instance, even at cruise, the wing on the Boeing 787 Dreamliner will nominally deflect ten feet (10% of the semispan) as shown in Figure 1. The weight savings offered by composite construction combined with active load alleviation and other modern flight controls point toward a future of ever more flexible transport aircraft. Both the "Double Bubble" D8[1] and truss-braced, wing "SUGAR"[2] concept aircraft (Figure 2) are designed around highly-flexible, high-aspect ratio composite wings. Beyond these, future concepts actively exploit wing flexibility through in-flight morphing to significantly improve performance throughout the mission flight profile allowing for even lighter construction. For example, the Variable Camber Continuous Trailing Edge Flap[3] (VCCTEF) concept depicted in Figure 3 can dynamically adjust the spanwise lift distribution to improve aerodynamic performance. These concepts make accurate static and dynamic aeroelastic analyses more critical than ever. Classic methods that assume small deflections are no longer valid and therefore new tools must be developed and deployed earlier in the design cycle.

Figure 1. Boeing 787 at cruise (courtesy of Boeing).

The proposed paper presents the first steps in the development of a static aeroelastic analysis capability that leverages the versatility and accuracy of Cart3D[4], an

* Senior Research Scientist, Advanced Supercomputing Division, MS 258-5; david.l.rodriguez@nasa.gov. Senior AIAA Member.
† Aerospace Engineer, Advanced Supercomputing Division, MS 258-5; michael.aftosmis@nasa.gov. Associate Fellow AIAA.
‡ Research Scientist, Advanced Supercomputing Division, MS 258-5; marian.nemec@nasa.gov. Senior AIAA Member.
§ Senior Aerodynamic Designer, Zee Aero. Associate Fellow AIAA.

Figure 2. Truss-braced wing concept.

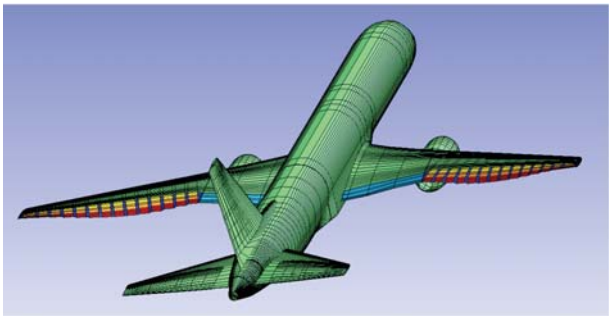

Figure 3. Representation of an aircraft using the Variable Camber Continuous Trailing Edge Flap concept.

inviscid Cartesian-mesh solver, and a mid-fidelity structural analysis model that idealizes the wing structure as a tapered beam. The combined analysis method is designed to handle discrete geometry whether it is a legacy surface mesh or derived from a modern computer-aided design software package. In the future, this capability will be leveraged in a multidisciplinary design optimization framework. Hence, the analysis is designed to properly represent the tradeoffs between aerodynamic performance and structural weight.

The aeroelastic analysis technique presented here is composed of three separate tools: an inviscid aerodynamic analysis method, a wing structures analysis method, and a geometry deformation tool. These three tools are coupled in a modular fashion to ultimately produce a system capable of analyzing very flexible wings operating at multiple flight conditions. This work is the first step in the development of an automated method which can be incorporated into a variable-fidelity, multidisciplinary design optimization framework.

II. Methodology

Aeroelastic analysis has been successfully performed by many by solving the fully-coupled set of aerodynamic flow and structural model equations. Drela[5] combined linear lifting-line and nonlinear beam theories to create a fully-coupled analysis method which used a global Newton method. Martins[6] solved not only solved a fully-coupled, high-fidelity aerostructural model but also the corresponding adjoint equations for use in design optimization frameworks. However, while solving the fully-coupled set of equations is usually feasible, it is not always practical. For high-fidelity and mixed-fidelity analysis methods, especially those with disparate solution schemes, a monolithic approach to solving the aerostructural problem is simply not flexible enough. The ability to use variable-fidelity analysis methods without the need to completely rewrite an existing analysis tool to include cross-discipline terms is a very attractive feature. Hence for the work in this proposed paper, the aerostructural problem is solved using a loosely-coupled methodology.

Figure 4 portrays a top-level overview of the approach. The baseline geometry is provided as a watertight triangulation of either the complete vehicle or its major components. This geometry feeds each part of the aeroelastic analysis in some manner. The triangulation is analyzed with the Cart3D simulation package to compute the initial loads on the baseline geometry. A structural model that is based on this outer mold line is then automatically built using the geometric analysis tools provided by the Blender[7] modeling suite. The computed loads are conservatively transferred to the structural model, which then predicts the deflected shape of the wing. Deformations from the structural model are applied to the wing geometry using an interpolating, 3D spatial deformer within Blender resulting in an updated geometry. This geometry is then automatically re-meshed for aerodynamic analysis and the process is repeated until the deformed shape converges. Once converged, the method provides a deformed version of the baseline geometry where the aerodynamic loads and the deflections predicted by the structures model are compatible. Details on each component of the analysis method are provided in the sections below.

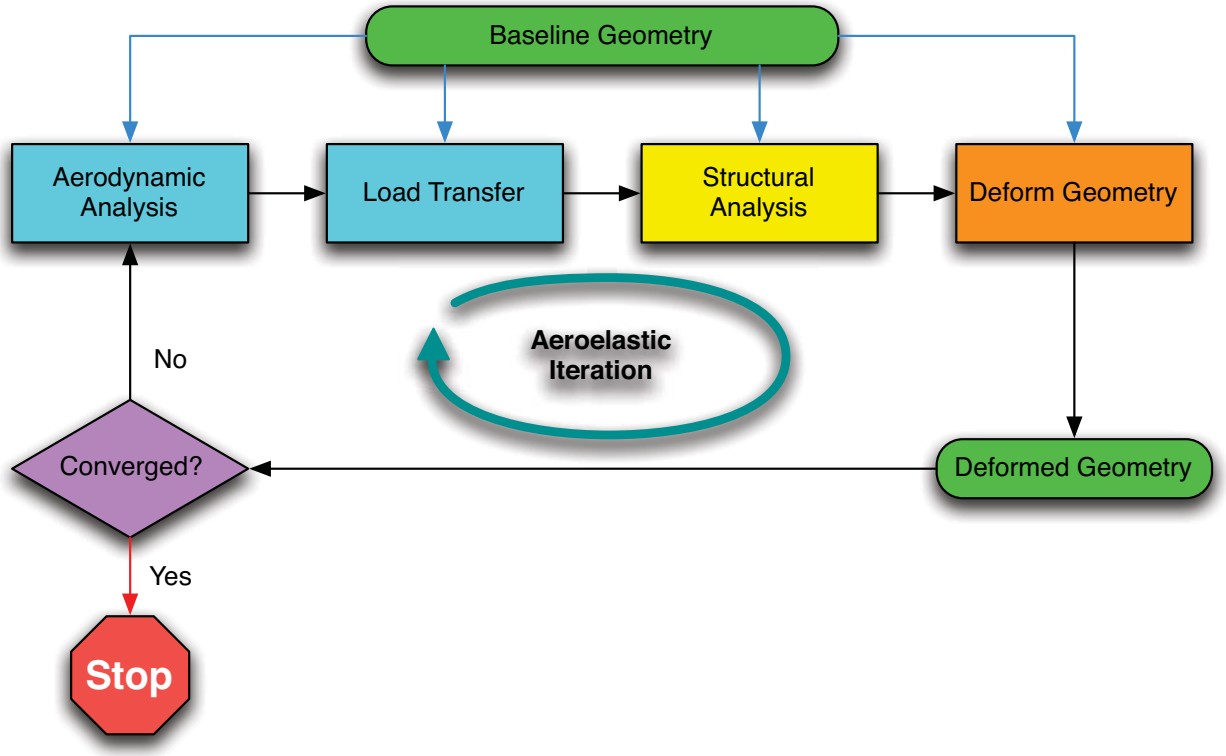

Figure 4. Aeroelastic analysis procedure. Blue connections indicate data that initially is computed and transferred only once, while black connections are repeated until convergence of the deformed wing shape.

A. Structural Analysis

This work employed a modified version of the structural analysis method originally developed by Gallman[8]. This mid-fidelity method models the wing structure as a tapered, closed box consisting of stringers and shear webs as shown in Figure 5. The stringers have a piecewise-constant cross-sectional area along the span and carry all of the bending loads. Figure 5 shows a model with six stringers but the middle stringers are optional. The shear webs are of piecewise-constant thickness along the span and carry all of the shear and torsional loads about the elastic axis of the model. Both the stringer cross-sectional area and the shear web thickness are user-specified and can vary on a per-element basis. Reference [8] performed relevant validation of this model through direct comparison with NASTRAN[9] results.

The structural model is discretized using a number of spanwise panels modeled as separate beam elements, each of which use a tapered-beam formulation. However, the elements are empirically modeled as tapered beam elements. Distributed loads (forces and moments) are applied locally to each element in the model. As shown in Figure 4, these loads are provided to the structural model, which predicts bending displacements along with local twist (torsional displacement) about the elastic axis. While lift-aligned bending accounts for most of the wing displacement, streamwise bending is important since it effectively modifies the wing sweep and thus affects longitudinal stability. Torsional displacement directly modifies the wing's local incidence angle and most directly affects the spanwise load distribution.

Since it computes two bending and one torsional displacement this is formally a three degree-of-freedom beam model. Compression or elongation along the beam axis is neglected and sectional yaw and roll rotations are

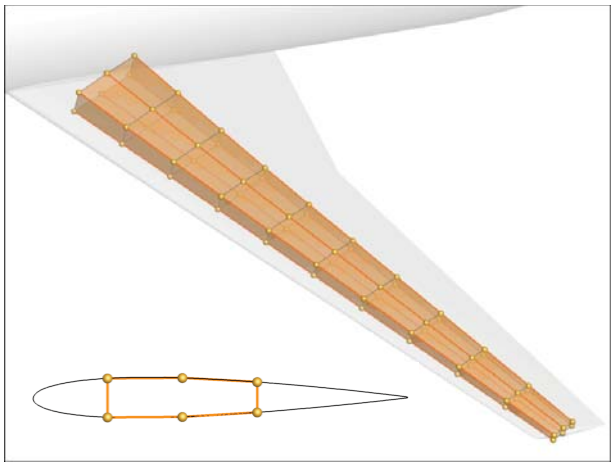

Figure 5. Wing structural model showing 6 constant area stringers (paneled between spheres) and 6 constant thickness shear webs.

assumed negligible. Like any finite-element model, accuracy is improved as the number of panels increases, and for the class of deformations considered here, mesh independent results are achieved with roughly 30 elements per semi-span. After computation, displacements from the structural model are provided to the deformation tool to produce a new surface triangulation.

This structural analysis model was chosen for two reasons. First it was originally developed to model a joined-wing structure meaning it is also applicable to a truss-braced wing such as that shown in Figure 2. This type of aircraft is currently of great interest at NASA research centers. The other reason is that the model is detailed enough for a design optimization framework. The spar areas and web thicknesses for each beam element can be optimized to provide ample structural stiffness at minimum weight. While the model is relatively simple, it still is directly tied to the outer-mold-line of the wing, which of course determines aerodynamic performance. This inherent coupling will provide an optimization framework with the proper tradeoffs between aerodynamic and structural efficiencies.

B. Aerodynamic Analysis

Aerodynamic analysis for all results in this paper is performed using the Cart3D simulation package which includes an adjoint-driven mesh refinement capability. The simulation package uses a Cartesian cut-cell approach[10] in which the governing equations are discretized on a multilevel Cartesian mesh with embedded boundaries. The mesh consists of regular Cartesian hexahedra everywhere, except for a layer of body-intersecting boundaries. The spatial discretization uses a second-order accurate finite volume method with a weak imposition of boundary conditions. The flux-vector splitting approach of van Leer[11] is used. Steady-state flow solutions are obtained using a five-stage Runge–Kutta scheme with local time stepping and multigrid. Domain decomposition via space-filling curves permits parallel computation; for more details see Aftosmis et al. and Berger et al.[12, 13, 14]

Although it consists of nested Cartesian cells, the mesh is viewed as an unstructured collection of control volumes making the approach well-suited for solution-adaptive mesh refinement. Mesh refinement is based on a duality-preserving discrete adjoint solver developed for Cart3D by Nemec et al.[15] This solver shares the same basic data structures, domain decomposition, and other infrastructure with the primal solver and achieves similar performance. While originally developed for gradient-based shape optimization[16], the method is also employed for output-based error-estimation and adaptive mesh refinement[17] referred to as the AERO module.

The adjoint-based error-estimation tailors the mesh refinement to reduce discretization error in a user-selected output of interest such as lift or drag. Error in a functional can be either driven below some pre-specified value, or alternatively, reduced as much as possible using a "worst-errors-first strategy" until attaining a desired mesh size. Adaptation is performed incrementally by cycling between the primal and adjoint solvers, with no more than one level of cell refinement implemented at a time. With this strategy, typical simulations cost 3-5 times that of a single flow solve on the final mesh. An example refined mesh and solution is portrayed in Figure 6.

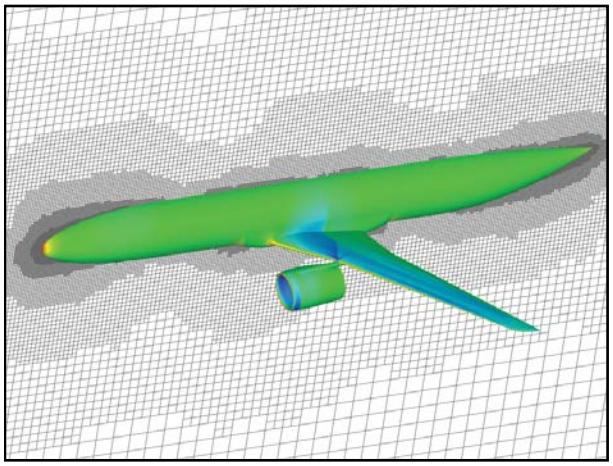

Figure 6. Example of a refined Cart3D mesh about a transport aircraft. The mesh refinement was driven by the accompanying adjoint solver. The surface coloring corresponds to local pressure.

Referring back to Figure 4, once a flow solution is computed, the surface pressure distribution is used to compute the distributed loads on the wing surface. These loads are then transferred to the structures model to compute deflections and therefore the deformed shape. More details on the transfer process are provided in the following section. Note the Cart3D solution also provides a prediction of the aerodynamic performance of the aircraft analyzed.

C. Loads Computation and Transfer

For most aeroelastic analyses and particularly when aerodynamic and structural analyses are loosely-coupled, the transfer of computed loads between the aerodynamic and structural analyses is not trivial. Aerodynamic loads computed on an aerodynamic surface mesh must be transferred to what is often a very different structural finite element model. How this is done is not always clear and often becomes somewhat arbitrary. For the work presented here however, the structural model idealizes the wing structure as a tapered wing box, which in turn is modeled as a cantilever beam with a user-specified number of spanwise elements. This greatly simplifies the process as only the

distribution of loads along the elastic axis is needed. To obtain this distribution, the surface triangulation is divided into regions that correspond to the elements of the structural model. Once identified, the triangles in these regions are then *explicitly bound* to the structural elements. This binding of the skin to the structural elements persists throughout the iterative process of the aeroelastic analysis. This is important since for cases with large deflections, the same regions of the wing surface must be used throughout the iterative process for consistency.

Since the load distribution along the elastic axis is required, the binding of the triangulation to the structural model is accomplished through the set of continuous strips whose boundaries are defined by the extended edges of the structural model elements. The strip boundaries are therefore mostly perpendicular to the elastic axis of the wing with two exceptions. Near the wing root, everything inboard of the outboard edge of the root structural panel is included in that element's binding. Similarly, at the wing tip, all of the surface outboard of the inboard edge of the tip panel is used to compute the load for that element. In other words, each triangle on the wing surface is bound to its nearest structural model panel. For a triangle whose area straddles more than one structural element, the centroid of that triangle is used to determine to which structural element it is bound. Figure 7 shows an example where the different colors indicate bindings to the 10 elements in the structural model.

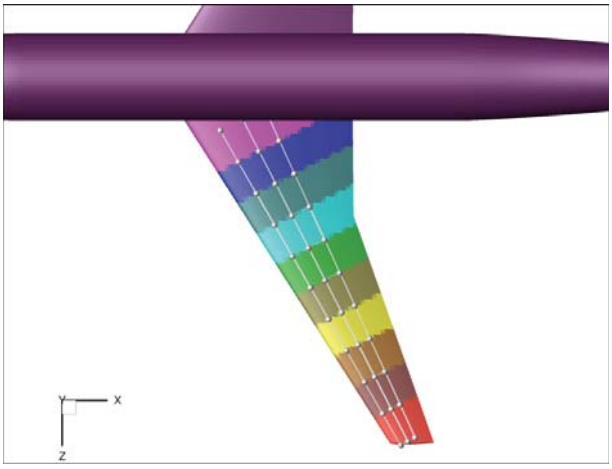

Figure 7. Example binding of surface triangles to each structural element as indicated by different colors on the wing.

The loads computed on each strip of triangles include the force component normal to and moment about the elastic axis. Forces parallel to the elastic axis are not considered as the structural model assumes no compression or elongation along the length of the beam. Also, the structural model currently ignores the local applied moment components that are not about the elastic axis as these are considered negligible. The validity of this assumption will be examined in the Applications section below. Referring back to Figure 4, the force and moment computed for each strip is applied to its corresponding beam element in the structural model. Note that as the wing deforms, the triangle binding remains fixed for consistency throughout the iterative computation. This means the binding only needs to be performed once, namely right after the structural model is built and before the aeroelastic iterations have begun.

D. Deformation Scheme

Blender[7] is a discrete geometry engine that was originally designed for computer graphics modeling and animation. Many of the deformation tools that are used in 3-D animation are also used in some form in modern shape optimization methods for aerospace designs. Blender also has an incredibly practical graphical-user-interface allowing the user to easily generate, modify, edit, and deform discrete representations of practically any geometric object. Since batch-mode manipulating and rendering is essential to 3-D animation, the package also provides a powerful Python-based[18] application-programming-interface (API) allowing users to develop their own extensions and scripts. The surface deformer for the aeroelastic analysis method is built using this API.

The specific Blender[5] tool that is used to deform the triangulation is the lattice modifier. An example of a lattice modifier is shown in Figures 8 and 9. The lattice is initially a three-dimensional, strictly Cartesian mesh in space which is bound to a discrete geometry. As the vertices (or points) of the lattice are moved, the bound discrete geometry is morphed through 3-D spline functions defined between lattice points in the three Cartesian directions aligned with the lattice. In the cases shown in Figures 8 and 9, the lattices are aligned so that the normals of the rectangular sections are perpendicular to the wing elastic axis. This means that displacements computed by the structural model are also the displacements of the lattice rectangle sections. Since the structural model does not predict deformations in the plane of any wing section (such as airfoil de-cambering), the lattice can be built simply as a spanwise array of rigid rectangles. In practice, each spanwise rectangle is always displaced in such a manner to maintain its original shape, though of course its location and orientation is allowed to vary. Also, since the rectangle dimension is only two in its own subspace, the spline function parallel to the rectangle is always effectively linear. Thus, the bound sectional shape is always preserved even as the wing spanwise shape is modified. This is important since in an actual transport wing structure, the ribs are usually nearly perpendicular to the wing elastic axis. As the wing bends, the ribs effectively preserve their shape. Conversely, in the direction parallel to the elastic axis, a cubic Hermite spline function (more specifically a Catmull-Rom spline) is used to smoothly deform the wing surface between the many lattice sections. Note that this spline is not currently in the Blender distribution and was added to the application by the author.

The translational and rotational displacements that are computed by the structures code are also discrete in that values are predicted for the centroids of the beam elements and the very ends of the wing box. These locations are not always convenient since the Blender lattice consists of equally spaced points. Therefore, to provide a continuous function of the displacements, the discrete values are splined with the traditional cubic spline that enforces continuous curvature. Figure 10 shows an example of computed discrete displacement values along with their corresponding splines.

The specific values that are splined are two components of beam deflection normal to elastic axis and the torsion-driven rotation about the wing elastic axis. While these are the only displacements that are computed by the structural model, displacements and rotations in the other directions can also be computed to preserve geometric integrity. For instance, as the wing tip is deflected vertically, the wing tip moves inboard parallel to the undeformed elastic axis to maintain the actual wing span. This is particularly necessary for wings exhibiting large deflections. Simply shearing the wing shape vertically would actually lengthen the wing span in terms of arc length and artificially increase the wing area. Similarly, as the wing bends, the local section roll angle (or dihedral) must remain perpendicular to the deformed elastic axis or the effective wing thickness is altered due to unrealistic shearing of the geometry. This is demonstrated in the example in Figure 8; notice that the lattice sections remain perpendicular to the wing elastic axis even as it bends. Likewise, as the wing is deflected in the streamwise direction, the lattice sections yaw to remain orthogonal to the elastic axis.

As discussed above, all six components of translation and rotation can be computed at any point along the span using the displacement splines computed by the structural model and geometric integrity constraints. Therefore, to apply the deformation with the lattice, the local displacements are computed for each spanwise lattice section according to the displacement splines and constraints. From there, a smooth deformed surface triangulation is generated that matches the deformed shape predicted by the structures model. This process is the last component of the iterative cycle in the Figure 4 architecture. The difference between the deformed geometries from the current and previous iterations vanishes with convergence of the coupled aerostructural system.

For a swept wing, the lattice often exists well into and even beyond the fuselage, as is the case in Figure 9. While only the wing part of the triangulation is bound to the lattice, Blender does not provide a natural way to ensure that the wing is not displaced at the intersection with the fuselage. To address this issue, a module was built for Blender that would gently fade the deflected wing geometry into the baseline geometry thereby preserving the intersection of the wing and fuselage. This is accomplished by "blending" the baseline and deflected wings with the weighting functions shown in Figure 11.

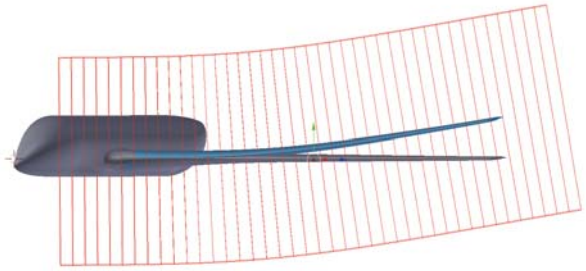

Figure 8. Example deformation of a wing with a Blender lattice modifier. Note the spanwise sections of the lattice are rolled to preserve orthogonality to the deflected wing and therefore the local thickness.

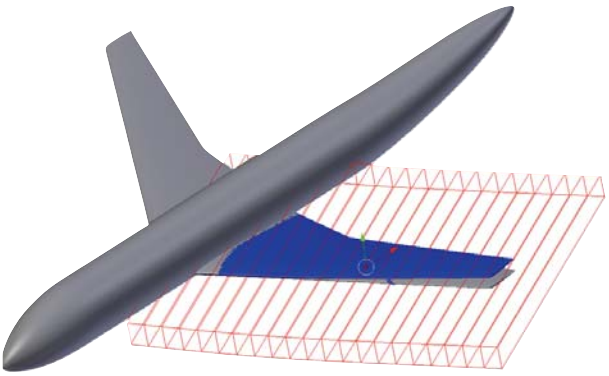

Figure 9. Example deformation of a wing with a Blender lattice modifier. Note the lattice is aligned with the elastic axis of the wing consistent with the structural model.

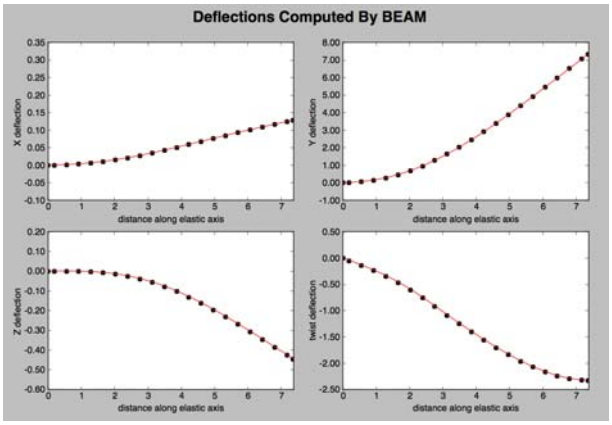

Figure 10. Example of splined displacements computed by structures model. Note that for these plots, the Z-axis is aligned with the elastic axis of the wing, and the X-axis is chordwise.

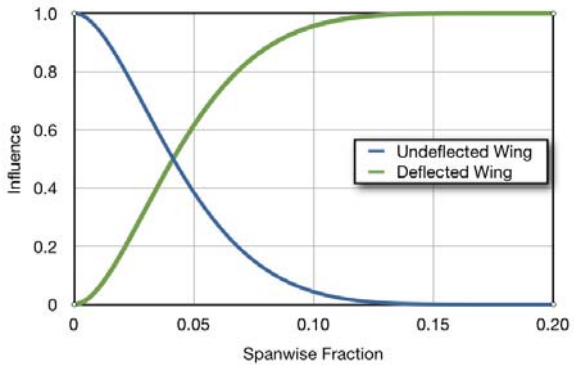

Figure 11. Weighting function used for blending baseline and deflected wings to maintain valid geometry near the root. Note the abscissa is the spanwise distance from the wing intersection with the fuselage.

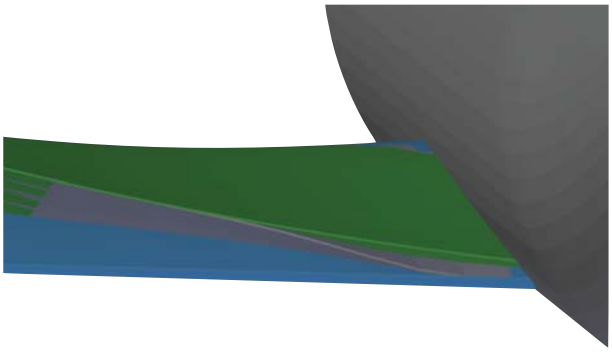

Figure 12. Example of baseline (blue), deflected (green), and blended (gray) wings portraying the effects of the weighting function in Figure 11. The view is upstream looking at the trailing edge with the fuselage on the right.

Note the functions are designed to always sum to unity. Each vertex on both wings is weighted by these functions based on the baseline vertex location in the spanwise direction (normal to the freestream direction in this case) and then summed to create a new wing. This wing follows the predicted deflection for most of its span, but remains constrained at the root section. This process not only maintains valid geometry, but it also better represents reality where the wing is clamped at the root. While in actuality the fuselage also deforms a small amount, this effect is not currently modeled. Clamping the wing at the root provides a realistic model for preserving the wing-fuselage function. The trailing edges of a baseline, deflected, and blended wing near the root of an example wing-fuselage configuration are shown in Figure 12. The blended wing by itself is shown in Figure 13 for clarity.

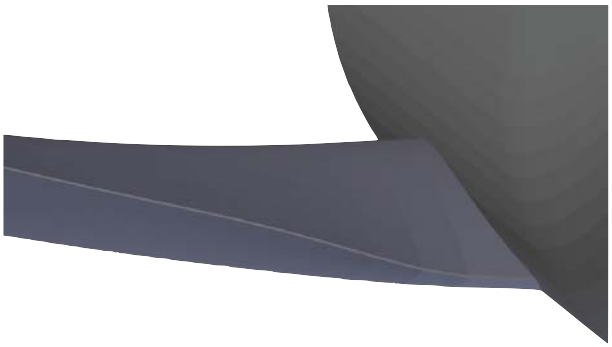

Figure 13. The blended wing from Figure 11.

E. Interface

The components all shown in Figure 4 and discussed in the sections above are all modular, stand-alone applications in themselves. To control the transfer of data between components and the overall execution of the method, an interface consisting of Python[18] modules and scripts has been created. Python was selected as the scripting language because the Blender API is also in Python and also for flexibility and ease of comprehension. The interface provides the user with the ability to execute the entire scheme in Figure 4 or just individual components to debug a problem setup without having to execute the entire process. The components that require the use of the Blender application can be executed in a manner where the results are presented within the Blender graphical-user-interface, thereby enhancing and accelerating the setup and debugging process. Other components produce several data files which can be viewed by many visualization packages. The interface also keeps track of convergence of the "aeroelastic iteration" (indicated in Figure 4) by outputting the wing tip deflection computed during each cycle. Using Python scripts and modules also allows the user to modify the process if necessary. This includes the ability to substitute, enhance, and even include other components or disciplines in the analysis.

More details of the interface will be provided in the final paper.

III. Applications

Preliminary runs of the aeroelastic analysis approach have been successfully completed on a generic wing-fuselage geometry shown in Figure 14. Note that this geometry is the same as that shown in Figures 5, 7, 9, and 10. The geometry was analyzed at a Mach number of 0.75, angle of attack of 3°, and altitude of 35,000 feet, which is a typical transonic cruise condition for a commercial transport. The structural model was somewhat flexible in that the final deflection at the wing tip was just over 5% of the semispan. The convergence history of this tip deflection dur-

Figure 14. Generic wing-fuselage geometry used for preliminary testing of aeroelastic analysis.

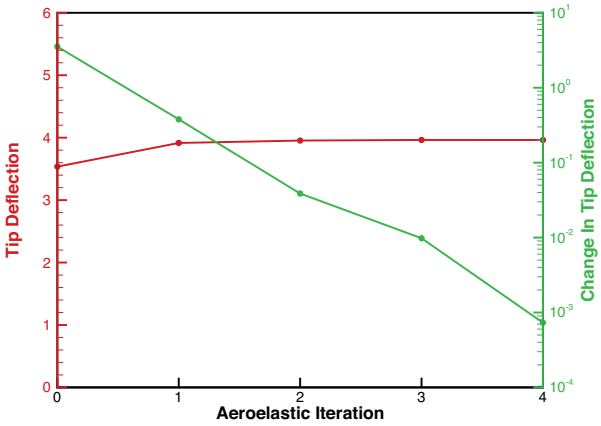

Figure 15. Convergence history of the tip deflection (in feet) computed during an aeroelastic analysis of a generic wing-fuselage geometry.

ing the analysis is given in Figure 15. A comparison of the baseline and bent wings is given in Figure 16. As shown by the green line in Figure 15, this particular analysis converged the wing tip deflection for the combined aerostructural system by about 3 orders of magnitude in 4 iterations. Note that an under-relaxation factor of 0.5 was applied uniformly in this case to improve convergence. Typically, on a swept-back wing and with no under-relaxation, the method will overshoot the correct deflection as the undeflected wing usually produces significantly more lift at the tip than the bent wing. The deflection of the swept wing will effectively increase the washout of the tip region, thereby reducing lift. This in turn reduces the amount of deflection and of course produces the overshoot which can be a detriment to convergence of the method. However, simply under-relaxing the predicted deflections at each iteration, the overshoot can be reduced or eliminated and the overall convergence properties of the scheme improved.

In terms of computational costs, the aerodynamic analysis required the bulk of the used computational resources. The structural model required a small fraction of the total time for initial set up (surveying the triangulation) but then ran practically instantaneously during the iterative process.

The computed forces on the bound triangular strips are presented in Table I. These forces are resolved in an orthogonal coordinate system aligned with the elastic beam axis. More specifically, $\mathbf{F_a}$ is aligned with the elastic axis, $\mathbf{F_n}$ with the flapping or lifting direction, and $\mathbf{F_\tau}$ is streamwise but normal to the elastic axis. Of course, the lift force is the greatest contributor by 1-2 orders of magnitude. Note that the other two forces are about the same order of magnitude. For streamwise bending, the force is not negligible since its effect is magnified by the large moment arm from the wing root. However, the force along the elastic axis can be considered negligible as it would contribute chiefly to compression or elongation. Hence, the inextensible beam assumption made in the structural model is indeed valid.

In the final paper, the aeroelastic analysis will be assessed for three cases. One of these geometries will be the wing used in the Aeroelastic Prediction Workshop[19], which will also provide a proper validation case for the scheme. While the disciplinary analysis methods may not be high enough in fidelity to accurately predict the results observed in the wind tunnel, it will still serve as a good "sanity check". Another geometry that will be ana-

Figure 16. Comparison of baseline (green) and deflected (red) wings produced by aeroelastic analysis.

Panel	F_τ	F_n	F_a
root	0.0052	0.1702	-0.0162
2	0.0026	0.1417	-0.0099
3	0.0000	0.1215	-0.0066
4	-0.0028	0.1042	-0.0053
5	-0.0033	0.0885	-0.0036
6	-0.0033	0.0730	-0.0025
7	-0.0028	0.0577	-0.0017
8	-0.0021	0.0420	-0.0013
9	-0.0012	0.0262	-0.0007
tip	0.0011	0.0113	0.0011

Table I. Computed forces (normalized by configuration lift) on example case. F_τ is streamwise but normal to the elastic axis, F_n is in the lifting direction, and F_a is along the elastic axis.

lyzed is the Common Research Model (CRM)[20] used in the 4th Drag Prediction Workshop[21], if only because it demonstrates a typical application of this method. Since the structures code also has the capability to analyze joined wings (in fact, that was its original intent), a truss-braced wing such as the SUGAR[1] concept shown in Figure 2 will also be analyzed as an example.

IV. Future Work

The aeroelastic analysis approach discussed here provides the opportunity for true, aeroelastic design optimization of typical transport and even truss-braced wings. The adjoint-driven, design capability of Cart3D provides a scheme for quickly computing aerodynamic performance sensitivities to geometric shape changes. The structural analysis was also selected to provide the capability to optimize the structure design itself. Working in tandem, the two analyses should properly represent the tradeoffs that exist in the design space of an aircraft wing, particularly for the very flexible wing designs of the future. Ultimately this method was created not just as a stand-alone analysis but more so to be exploited within a design optimization framework. While the exact architecture of such a framework has not been established, the development of the framework is currently planned for the near future.

More details on future planned and potential work will be presented in the final paper.

V. Conclusions

Proper conclusions will presented in the final paper based on the performance of the method on the various test cases.

VI. Acknowledgments

Proper acknowledgements will be presented in the final paper, including the funding organizations and others who have contributed to the work in some way.

References

[1]Drela,M., "Development of the D8 Transport Configuration," AIAA 2011–3970, June 2011.

[2]Bradley, M. K. and Droney, C. K., "Subsonic Ultra Green Aircraft Research Phase II: N+4 Advanced Concept Development," NASA CR-2012-217556, May 2012.

[3]Urnes, J., Nguyen, N., Ippolito, C., Totah, J., Trihn, E., and Ting, E., "A Mission-Adaptive Variable Camber Flap Control System to Optimize High Lift and Cruise Lift-to-Drag Ratios of Future N+3 Transport Aircraft," AIAA 2013-0214, January 2013.

[4]Aftosmis, M. J., Berger, M. J., and Adomavicius, G., "A Parallel Multilevel Method for Adaptively Refined Cartesian Grids with Embedded Boundaries," AIAA 2000-0808, January 2000.

[5]Drela, M., "Method for Simultaneous Wing Aerodynamic and Structural Load Predictions," AIAA-89-2200, June 1989.

[6]Martins, J. R. R. A., *A Coupled-Adjoint Method for High-Fidelity Aero-Structural Optimization*, PhD thesis, Stanford University, Stanford, CA, 2002.

[7]http://www.blender.org

[8]Gallman, J. W. and Kroo, I. M., "Structural Optimization for Joined-Wing Synthesis," Journal of Aircraft, Vol. 33, No. 1, January-February 1996.

[9]Rodden, William P. Author: Johnson, Erwin H., "MSC/NASTRAN aeroelastic analysis: User's guide, version 68." MacNeal-Schwendler Corp., Los Angeles 90041-1777, 1994

[10]Aftosmis, M. J., Berger, M. J., and Melton, J. E., "Robust and efficient Cartesian mesh generation on component based geometry," AIAA Journal, Vol. 36, No. 6, June 1998, pp. 952–960.

[11]van Leer, B., "Flux-Vector Splitting for the Euler Equations," ICASE Report 82-30, September 1982.

[12]Aftosmis, M. J., Berger, M. J., and Adomavicius, G. D., "A parallel multilevel method for adaptively refined Cartesian grids with embedded boundaries," AIAA 2000-0808, January 2000.

[13]Aftosmis, M. J., Berger, M. J., and Murman, S. M., "Applications of Space-Filling-Curves to Cartesian methods in CFD," AIAA 2004-1232, January 2004.

[14]Berger, M. J., Aftosmis, M. J., and Murman, S. M., "Analysis of Slope Limiters on Irregular Grids," AIAA 2005−0490, January 2005.

[15]Nemec, M., Aftosmis, M. J., Murman, S. M., and Pulliam, T. H., "Adjoint Formulation for an Embedded-Boundary Cartesian Method," AIAA 2005-0877, January 2005.

[16]Nemec, M. and Aftosmis, M. J., "Adjoint Sensitivity Computations for an Embedded-Boundary Cartesian Mesh Method," Journal of Computational Physics, Vol. 227, 2008, pp. 2724-2742.

[17]Nemec, M. and Aftosmis, M. J., "Adjoint Error-Estimation and Adaptive Refinement for Embedded-Boundary Cartesian Meshes," AIAA 2007–4187, June 2007.

[18]http://python.org

[19]Heeg, J., Chwalowski, P., Florance, J. P., Wieseman, C. D., Schuster, D. M., and Perry, B., "Overview of the Aeroelastic Prediction Workshop," AIAA 2013-0783, January 2013.

[20]Vassberg, J. C., DeHaan, M. A., Rivers, S. M., and Wahls, R. A., "Development of a Common Research Model for Applied CFD Validation Studies," AIAA 2008-6919, August 2008.

[21]Vassberg, J. C., Tinoco, E. N., Mani, M., Rider, B., Zickuhr, T., Levy, D. W., Broderson, O. P., Eisfeld, B., Crippa, S., Wahls, R. A., Morrison, J. H., Mavriplis, D. J., Murayama, M., "Summary of the Fourth AIAA CFD Drag Prediction Workshop," AIAA 2010-4547, June 2010.